

Experimental Results Towards Content-Based Sub-Image Retrieval

Tao Wang

Juhua Shi

Mario A. Nascimento

*Department of Computing Science
University of Alberta, Canada
{trysi, juhua, mn}@cs.ualberta.ca*

Abstract

In this paper we are interested in the problem of sub-image retrieval (CBSIR), i.e., given a query image one must find the best candidate images that contain that query image. We used two kinds of image feature vectors: global color histograms and autocorrelograms and experimented with several distance measures for both feature vectors in our experimental system. After extensive experimentation we found that using autocorrelograms with the so-called S_1 distance measure yielded excellent results for sub-image retrieval with an acceptable processing overhead.

Keywords: Content-based sub-image retrieval (CBSIR), distance measure, image databases, GCHs (global color histograms), autocorrelograms

1. Introduction

Even though there have been standard retrieval techniques for text [2], they are not suitable for image data since image annotation is unfeasible for any non-trivial scenario. It is much more difficult to retrieve image data than text due to the subjectivity of human perception, which is difficult to represent formally. The latter problem is called Content-Based Image Retrieval (CBIR).

Given the difficulty of finding feature vectors to capture image information as comprehensively as human perception, most image retrieval approaches use the image's color feature since it conveys the basic information of an image. Global Color Histograms (GCHs) are very popular because they are simple and intuitive [3, 4, and 5]. As well, most importantly, they are efficient and insensitive to small changes in viewpoint [6]. However, a color histogram provides only the global distribution of color, i.e., images with similar histograms may have quite different content.

The Color Coherent Vector (CCV) method uses a histogram-refinement approach [4, 7]. The technique

classifies each pixel in a color bucket as either coherent or incoherent, depending upon whether the pixel is a constituent of a large similarly-colored region. The argument is that the comparison of coherent and incoherent feature vectors between two images allows for a finer distinction of similarity than when using GCHs.

The color correlograms [8] approach proposes a new use of the color feature for image indexing/retrieval. It expresses how the spatial correlation of pairs of colors changes with distance. The results reported in [8] show that using color correlograms may outperform both the traditional GCH method and the CCV method for image retrieval. Results also indicate that the autocorrelograms (simplified color correlograms) also achieved good results with reasonable trade-off.

There are a number of ways for improving a content-based image retrieval system, such as exploring more effective features to represent an image, choosing more accurate distance measures, finding better ways to evaluate the performance of the image retrieval system etc. The survey presented in [9] presents state-of-the-art technique in CBIR, including research related to the use of visual features (e.g., colors, textures, shapes, faces, edges, etc). To our knowledge not much research has been on done on sub-image retrieval, i.e., the task of find image(s) that *contain* a query image. Note that this is very different from the usual CBIR task, i.e., the retrieval of images which are similar (as a whole) to the query image.

Our objective in this paper is to find an effective way to perform content-based sub-image retrieval (CBSIR) and ranking [10, 13]. In our experimental approach, we used both GCHs and autocorrelograms. For distance measures we used the L_1 , D_1 and S_1 distance measures. Details about the distance measures are given in Section 2. To evaluate the performance of the image retrieval system, we chose four kinds of measures, as detailed in Section 3, which also reports the obtained results. Section 4 concludes the paper, and an Appendix present some sample queries and ranks yielded by the various approaches we have experimented with.

2. Our Approach for CBSIR

Figure 1 shows the framework of our CBSIR (Content-based Sub-Image Retrieval) system.

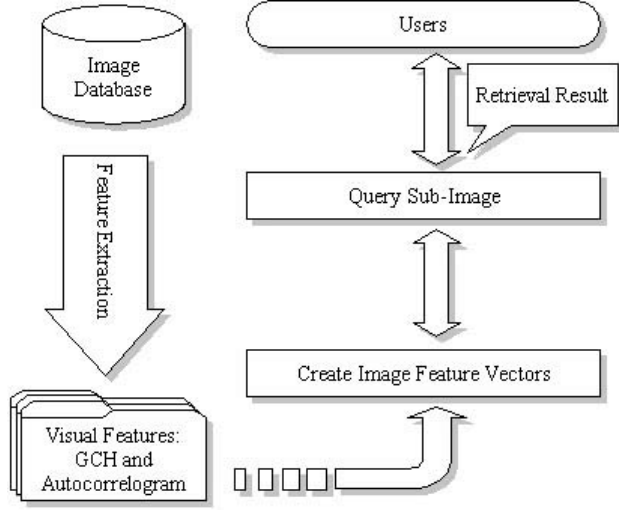


Figure 1. CBSIR (Content-based Sub-Image Retrieval) System Framework

2.1 Feature Extraction

As outlined above, in our approach we use the GCHs (Global Color Histograms) feature and the simplified color correlograms feature (autocorrelograms), both of which are extracted from an image using different models.

Definitions/Assumptions:

1. Let I be an image of size $n_1 \times n_2$ pixels.
2. The colors space in an image I is quantized into m colors c_1, \dots, c_m .
3. For a pixel $p = (x, y) \in I$, and $C(p)$ denotes its color, we define $I_c = \{p | C(p) = c\}$.
4. The notation $p \in I_c$ is synonymous with $p \in I$, and $C(p) = c$.
5. Let us denote the set $\{1, 2, \dots, m\}$ by $[m]$. We denote the spatial distance between pixels $p_1 = (x_1, y_1)$ and $p_2 = (x_2, y_2)$, using the L_∞ norm, i.e., $|p_1 - p_2| = \max\{|x_1 - x_2|, |y_1 - y_2|\}$.

GCH (Global Color Histogram) Feature.

Assuming an m -color model, a GCH is an m -dimensional feature vector (h_1, h_2, \dots, h_m) , in which each h_i represents the (usually) percentage of color pixels in an image corresponding to each color c_i .

The global color histogram h of image I is defined as the following:

For any pixel p from image I , $h_{c_i}(I)$ gives the probability that the color of pixel p is c_i :

$$h_{c_i}(I) = \text{probability}[p \in I_{c_i}] = \frac{\|I_{c_i}\|}{n_1 * n_2}$$

where $\|I_{c_i}\|$ is the number of pixels of color c_i in I .

Autocorrelogram Feature.

The feature gives the probability that a pixel at a distance k away from the given pixel is of color c_j . In this paper we use a simpler color correlogram feature, the autocorrelogram, which is reported [8] to yield good results with acceptable performance.

Let a distance d be fixed a priori. The correlogram of I is defined for $i, j \in [m], k \in [d]$ as:

$$\gamma_{c_i, c_j}^{(k)}(I) = \text{probability}[p_2 \in I_{c_j} \mid p_1 - p_2 = k]_{p_1 \in I_{c_i}, p_2 \in I}$$

The autocorrelogram of image I , which captures only the spatial correlation between identical colors, is defined as above but with $i = j$, i.e:

$$\gamma_{c_i, c_i}^{(k)}(I) = \text{probability}[p_2 \in I_{c_i} \mid p_1 - p_2 = k]_{p_1 \in I_{c_i}, p_2 \in I}$$

For each image in a database, we compute GCH and Autocorrelogram (a specification of color correlogram) feature vectors and store them in the feature database.

2.2 Distance Measures

We compute three distance measures, L_1 , D_1 and S_1 , for both GCHs and autocorrelograms in our experimental system. While the L_1 and D_1 distance measures were also used elsewhere, e.g., [1], we believe that the S_1 distance measure is being used for the first time as a distance measure for autocorrelograms. All the distance measures are defined and computed as below.

L_1 Distance Measure [8]

The GCH L_1 distance is the sum of color histogram differences, bin-by-bin, between two images. The formula to compute the GCH L_1 distance is:

$$|I - I'|_{h, L_1} = \sum_{i \in [m]} |h_{c_i}(I) - h_{c_i}(I')|$$

The autocorrelogram L_1 distance is the sum of autocorrelogram differences between two images, color-by-color. The formula to compute autocorrelogram L_1 distance is:

$$|I - I'|_{\gamma, L_1} = \sum_{i \in [m], k \in [d]} |\gamma_{c_i, c_i}^{(k)}(I) - \gamma_{c_i, c_i}^{(k)}(I')|$$

D_1 Distance Measure [8]

The GCH D_1 distance is the sum of normalized color histogram difference between two images. The formula to compute the GCH D_1 distance is:

$$|I - I'|_{h,d_1} = \sum_{i \in [m]} \frac{|h_{c_i}(I) - h_{c_i}(I')|}{1 + h_{c_i}(I) + h_{c_i}(I')}$$

Similarly the autocorrelogram D_1 distance is defined as:

$$|I - I'|_{\gamma,d_1} = \sum_{i \in [m], k \in [d]} \frac{|\gamma_{c_i,c_i}^{(k)}(I) - \gamma_{c_i,c_i}^{(k)}(I')|}{1 + \gamma_{c_i,c_i}^{(k)}(I) + \gamma_{c_i,c_i}^{(k)}(I')}$$

S₁ Distance Measure.

The retrieval performance of a CBSIR System is determined both by the quality of the features used to represent the image content and by appropriate distance measure. Distance measure is to reflect the perceptual

$$|I - I'|_{h,S_1} = 1 - \frac{1}{m} * \sum_{i \in [m]} \frac{\min(h_{c_i}(I), h_{c_i}(I'))}{\max(h_{c_i}(I), h_{c_i}(I'))}$$

similarity of two images. With that in mind, we designed the following S_1 distance:

The formulation is inspired, but different from the following one, used in [12]:

$$|I - I'|_h = 1 - \sum_{i \in [m]} \min(h_{c_i}(I), h_{c_i}(I'))$$

Consider the following example. Given three images I_1 , I_2 and I_3 , suppose we choose the color space as $m=3$, the probability of each of color (histogram bins) is the following:

	Red	Green	Blue
Image I_1	0.3	0.2	0.5
Image I_2	0.6	0.2	0.2
Image I_3	0.5	0.3	0.2

By using the equation from [12], we would obtain:

$$|I_1 - I_2|_h = 1 - (0.3 + 0.2 + 0.2) = 0.3$$

$$|I_1 - I_3|_h = 1 - (0.3 + 0.2 + 0.2) = 0.3$$

It shows the same difference between I_1 and I_2 and between I_1 and I_3 since it treats the contribution of different colors to the image dissimilarity equally. We replaced $\min(h_{c_i}(I), h_{c_i}(I'))$ by $\frac{\min(h_{c_i}(I), h_{c_i}(I'))}{\max(h_{c_i}(I), h_{c_i}(I'))}$ in order to emphasize the contribution of colors which have very different distributions between the images.

The S_1 distances between image pairs now become:

$$|I_1 - I_2|_{h,S_1} = 1 - (0.3/0.6 + 0.2/0.2 + 0.2/0.5)/3 = 0.37$$

$$|I_1 - I_3|_{h,S_1} = 1 - (0.3/0.5 + 0.2/0.3 + 0.2/0.5)/3 = 0.44$$

The distances reflect better the similarity between I_1 , I_2 and I_3 .

$$|I - I'|_{h,S_1} = 1 - \frac{1}{m} * \sum_{i \in [m]} \frac{\min(h_{c_i}(I), h_{c_i}(I'))}{\max(h_{c_i}(I), h_{c_i}(I'))}$$

The Autocorrelogram S_1 distance is similarly defined as:

In this paper we use L_1 and D_1 distance measure for two reasons: first they are commonly used in feature comparison, another reason is that we want to compare our CBSIR retrieval results with the results in Huang's dissertation [1], so we use same distance measures. In addition, we use the *autocorrelogram* S_1 distance measure, which is different from Huang's autocorrelogram approach, and as far we know is a novel combination of feature vector and distance measure.

2.3 Efficiency Consideration

Memory cost

If we choose a color space of 64 colors, each image will need 64 bytes to store GCH feature vectors. We use distances 2, 5 and 8 for the autocorrelogram feature vectors and the same color space as for the GCH, hence the autocorrelogram feature vectors will cost 192 bytes for each image, three times larger than GCH's memory cost.

Query Time cost

For each image to be queried, there are two parts of time costs in our approach.

The first part is feature vector computing cost. GCH needs to check every pixel color value in the image; while for each pixel in the image, autocorrelogram method needs check all the pixels with a certain distance. Since our experimental system uses 2-pixel, 5-pixel and 8-pixel distances to compute autocorrelogram, we need check 120 (16+40+64) pixels' color value for each pixel in the image. Autocorrelogram may take as much as 120 times cost to compute feature vector as GCH.

The second part is distance measure computing and ranking computing cost.

We need to compute the distance for each image in the database. If we let the image database size be N , the GCH's cost in computing distance measure is $O(N)$. Since we chose three distances (2, 5 and 8 pixels) in autocorrelogram method, we expect autocorrelogram will take about three times as much as GCH's. For the ranking computing, it is only relevant to the image database size. And we know it is $O(N \log N)$. It is easy to find that first

part cost is irrelevant to the image database size and the second part cost will grow as the image database size grows. For very small image database (which is unlikely to happen), the total query cost of autocorrelogram may take 120 times as much as GCH's. For larger image database, the second part cost will be more significant. The total query cost will be the same for autocorrelogram and GCH if N is close to infinite.

3. Experimental System

In order to evaluate the results obtained when processing a query, we use average r -measure and average $p1$ -measure [5, 10] to evaluate the effectiveness of our schemes using the following measures:

- *r-measure*: sum, over all queries, of the rank of the correct answer
- *average r-measure*: r -measure divided by the number of queries
- *p1-measure*: the sum of the reciprocal of the rank of all queries
- *average p1-measure*: $p1$ -measure divided by the number of queries

As we shall discuss shortly we also complement the measures above with the system's memory and time requirements.

3.1 Establish Testbed

To guide the research effort in the correct direction, evaluating the system performance is important. First of all, we need to establish a well-balanced large-scale testbed. It has to be large in scale to test the scalability and balanced between image content to the test image feature's effectiveness and the system's overall performance [9].

Hence, we use an image database consisting of 5,200 color JPEG images, of sizes 900 x 600 and 600 x 900. The content of images ranges from animals, ocean, clouds, space and sand to human beings.

Let $\{Q_1, Q_2, \dots, Q_q\}$ be the query images. We create the query images set by cropping sub-images from 21 randomly selected images from testbed image database. The original images are our desired query results.

There are two parts to build a Content-Based Sub-Image Retrieval (CBSIR) system:

1. Generating the feature vectors database (both GCH and autocorrelogram). We compute feature space vectors for all the images in the testbed and store the vectors in two files. One is the feature file for the GCH vector and the other is the feature file for the autocorrelogram vector.

2. Image query system. We compute the feature vectors of the objective image and find the most similar images from the image database (testbed) using the four distance measures discussed earlier.

The features vector database only needs to be generated once, while the image query system will be used repeatedly. Hence, we will focus on Part 2 when we analyze the system's costs.

3.2 Experimental Results & Analysis

The following performance results were collected under Red Hat 7.0 running on a computer with a PII- 350 MHz CPU and 128Mb of main memory.

We randomly chose 21 images from the testbed. As mentioned earlier, we evaluate the performance of our CBSIR (Content-Based Sub-Image Retrieval) system using two criteria. One is the average r -measure and $p1$ -measure to reflect the effectiveness of our schemes. The other is the space and time cost of the system to evaluate the efficiency of our schemes.

In querying sub-images, we need to compute the objective image feature vectors in order to find the least distance images from the image database. Four different distance measures with four kinds of system performance measure are tested. We use average r -measure and average $p1$ -measure to evaluate the effectiveness of our schemes. The smaller the r -measure is, the better the system performance. The larger the $p1$ -measure is, the better the system performance.

Time cost of the CBSIR system

Table 1 shows the time cost (measure in seconds) of querying an image using different feature vectors and all distances explained earlier.

Time Cost	GCH L_1	GCH D_1	GCH S_1	Auto L_1	Auto D_1	Auto S_1
Max	0.02	0.03	0.03	0.15	0.16	0.17
Min	0.01	0.02	0.02	0.12	0.14	0.14
Avg.	0.01	0.02	0.03	0.14	0.15	0.16

Table 1. Time cost for the different measures (sec)

We pointed out earlier that the distance measure based on autocorrelogram will cost from 1 to 120 times as much time as the distance measure based on GCH. The data collected from our experimental system were consistent with what we expected from our analysis.

Effectiveness of the CBSIR system

Table 2 shows the p-measures and r-measures (accumulated over 21 queries) as well as the average values.

Different Measure	r-measure	average r-measure	p1-measure	average p1-measure
Auto L_1	417	19.86	16.41	0.78
Auto D_1	2474	117.81	13.17	0.63
Auto S_1	31	1.48	18.96	0.9
GCH D_1	8359	398.05	10.74	0.51
GCH L_1	84	4	15.57	0.74
GCH S_1	9731	463.38	11.48	0.55

Table 2. Summary of obtained results

We know that the smaller the r-measure and the greater the p-measure, the more effective the system is. We can then conclude that autocorrelogram S_1 distance outperformed all the other distance or vector measures in both p-measure and r-measure criteria.

The memory needs for this program are in two parts: the memory for the program, which is quite small, being less than 1Mb in our experimental system; and the memory for the data, which should be about the size of GCH and correlogram feature base files, it is about 10M in our experimental system.

4. Conclusion

Sub-image retrieval is an interesting and challenging topic. In this paper, we tested GCH and autocorrelogram feature with L_1 and D_1 distance measure in sub-image retrieval. We also proposed a more effective distance measure S_1 . Based on the sample queries and results of p-measure and r-measure, our CBSIR system is effective in sub-image search and retrieval.

We also obtained results for CPU and memory costs in our CBSIR system and believe that, given the effectiveness of our results, the overall cost is acceptable.

Future work should be directed towards using larger databases as well as other images features, e.g., texture.

Acknowledgments




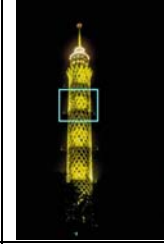


This research was partially supported by NSERC Canada.



References



- [1] J. Huang, Ph.D. thesis, *Color-Spatial image indexing and applications*, Cornell University, 1998
- [2] B. Ribeiro-Neto, R. Baeza-Yates, *Modern Information Retrieval*, Addison Wesley, 1999
- [3] W. Niblack et al, "The QBIC Project: Querying Images by Content, Using Color, Texture, and Shape", *Storage and Retrieval for Image and Video Databases (SPIE 1908)*, 1993, pages 173-187.
- [4] G. Pass, R. Zabih, and J. Miller, "Comparing images using color coherence vectors", *Fourth ACM International Multimedia Conference*, pages 65-73, 1996.
- [5] J. R. Smith and S.-F. Chang, "Tools and techniques for color image retrieval", *Storage and Retrieval for Image and Video Databases IV (SPIE 2670)*, 1996, pages 426-437.
- [6] M. Swain and D. Ballard, "Color indexing", *International Journal of Computer Vision*, 1991
- [7] G. Pass and R. Zabih, "Histogram refinement for content-based image retrieval", *IEEE Workshop on Applications of Computer Vision*, 1996, pages 96-102
- [8] J. Huang, S. R. Kumar, M. Mitra, W.-J. Zhu and R. Zabih. "Image Indexing Using Color correlograms", *IEEE Computer Vision and Pattern Recognition Conference*. San Juan, Puerto Rico, June 1997.
- [9] Y. Rui, T. S. Huang, and S.-F. Chang, "Image Retrieval: Past, Present, and Future", *Intl. Symposium on Multimedia Information Processing*, Dec. 1997.
- [10] W. Hsu, T. S. Chua, and H. K. Pung, "An integrated color-spatial approach to content based image retrieval", *3rd ACM Intl. Multimedia Conf.*, 1995, pages 305-313.
- [12] M. Ortega, Y. Rui, K. Chakrabarti, S. Mehrotra, T. S. Huang, "Supporting Similarity Queries in MARS", *5th ACM Intl. Multimedia Conf.*, 1997, pages 403-413.




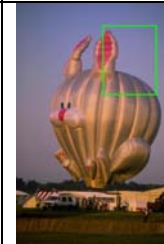


Appendix

Sample of results – The sub-images marked by the rectangle were used as queries and the numbers denote the position (rank) the whole image was returned by each combination feature vector/distance measure.

Ranks of different methods						
Auto L1	12	1328	160	2	7	371
Auto D1	646	2375	332	79	577	1728
Auto S1	1	1	10	1	1	3
GCH L1	80	3	19	474	12	2
GCH D1	2046	2258	70	315	311	1004
GCH S1	175	33	2	68	4	2

Ranks of different methods				
Auto L1	115	1	2	132
Auto D1	485	30	2	159
Auto S1	5	1	2	3
GCH L1	15	446	867	8
GCH D1	242	925	2	206
GCH S1	9	940	44	6

Ranks of different methods				
Auto L1	5	15	1	5
Auto D1	60	2	1	42
Auto S1	5	3	1	1
GCH L1	18	24	16	48
GCH D1	130	3	4	81
GCH S1	28	30	6	147

Ranks of different methods						
Auto L1	125	1	1	2	1	1
Auto D1	148	1	14	6	1	3
Auto S1	1	7	1	1	1	1
GCH L1	16	1213	629	12	3	12
GCH D1	104	16	1	1	1	1
GCH S1	1	316	2	1	5	1